

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Claims 1-5 (Canceled)

6. (Currently amended) A method of emulating a ~~handheld~~-video game device of the type that includes a ~~handheld~~-housing having an electronic display thereon, said ~~handheld~~-housing having therein a processor that runs a video game software image out of a page-based read only memory (ROM) to present interactive displays on said electronic display of animated video game play in response to user inputs, the method comprising:

executing a video game device emulator program on a target computing device different from said ~~handheld~~-video game device including said ~~handheld~~ housing having said electronic display thereon, said target computing device being capable of displaying graphical information on a target computing device display, said target computing device having read/write memory and receiving user inputs, said executing video game device emulator program controlling said target computing device to at least in part emulate said ~~handheld~~-video game device so as to at least in part enable said target computing device to run said video game software and present interactive displays of said animated video game play on said target computing device display in response to user inputs to said target computing platform;

modeling at least some display timing activities of said ~~handheld~~-video game device electronic display on said target computing device;

processing, with said emulator program executing on said target computing device, said video game software image capable of being executed on said ~~handheld~~-video game device processor within said ~~handheld~~-housing that runs video game software to present interactive displays on said electronic display of animated video game play in response to user inputs; and

generating a real time interactive video game presentation on said target computing device display at least in part in response to said processed video game software image and said modeled display timing activities,

wherein said video game software image comprises multiple ROM pages and said method further includes said emulator program (a) providing a pointer table system that allocates emulated ROM pages in said target computing device read/write memory to emulate said multiple ROM pages of said video game software image, and ~~duplicates~~ (b) duplicating at least a portion of one of said multiple ROM pages into plural of said allocated emulated ROM pages ~~across said ROM pages~~ to facilitate page selection and reduce page swapping, and

wherein the method further comprises using said pointer table system to control memory access by remapping native memory access instructions into function calls.

Claims 7-16 (Canceled)

17. (Previously presented) The method of claim 6 wherein said target computing device display comprises a liquid crystal display.

18. (Currently amended) The method of claim 6 wherein said ~~handheld~~ display comprises a liquid crystal display and said modeling comprises modeling a virtual liquid crystal display controller state machine corresponding to said ~~handheld~~-liquid crystal display to maintain real time synchronization with events as they would occur on said ~~handheld~~-video game device.

19. (Previously presented) The method of claim 6 further including using hardware-assisted BLIT memory transfer operations to efficiently transfer graphics information.

20. (Previously presented) The method of claim 6 further including using a pre-computed translation table that translates native platform graphics character formats.

21. (Currently amended) The method of claim 6 further including emulating registers and hardware-based memory structures within the ~~handheld~~-video game device in random access memory under software control.

22. (Previously presented) The method of claim 6 further including using a jump table to efficiently parse incoming binary instruction formats.

23. (Previously presented) The method of claim 6 further including using said pointer table system to control memory access by remapping memory access instructions into function calls.

24. (Previously presented) The method of claim 6 further including using said pointer table system to implement a read only memory protection function to eliminate overwriting of read only memory.

25. (Previously presented) The method of claim 6 wherein said modeling includes using a state machine defining at least a horizontal blank state and a vertical blank state.

Claim 26 (Canceled)

27. (Previously presented) The method of claim 6 further including selectively skipping frames while maintaining execution of instructions to maintain state information while minimizing game play slowdowns.

28. (Previously presented) The method of claim 6 further including performing no-operation look-ahead to avoid wasting processing time in no-operation loops.

29. (Currently amended) The method of claim 6 further including modeling each ~~handheld~~-video game device native instruction register as a union of byte, word and long register formats.

30. (Currently amended) The method of claim 6 further including modeling ~~handheld~~-video game device native instruction CPU flags to allow efficient updating after operations are performed by the target computing device.

31. (Currently amended) The method of claim 6 further including mapping the ~~handheld~~-video game device emulated program counter into at least one target computing device microprocessor general purpose register.

32. (Previously presented) The method of claim 6 further including providing an adaptable input controller emulator to receive user inputs from a variety of different user input devices.

33. (Previously presented) The method of claim 6 further including using screen memory buffers larger than said predetermined display area to increase paging efficiency by eliminating clipping calculations and using hardware Bitblit to transfer a subset of said memory buffer to display video memory.

34. (Previously presented) The method of claim 6 wherein said target computing device comprises an airline seat back controller.

35. (Previously presented) The method of claim 6 wherein said target computing device comprises a personal digital assistant (PDA).

36. (Currently amended) The method of claim 6 wherein said target computing device comprises a ~~handheld~~ portable computing device.

Claim 37 (Canceled)

38. (Currently amended) The method of claim 6 further including using said allocated emulated ROM pages to emulate read only memory arrays within a ROM-based pluggable game cartridge.

39. (Currently amended) The method of claim 6 further including allocating, in random access memory, at least twice the space occupied by ROM pages in the ~~handheld~~-video game device, and duplicating half of each page allocated in random access memory.

40. (Currently amended) The method of claim 6 wherein said ~~handheld~~ video game device is adapted for use with a pluggable game cartridge ROM having ROM banks, and said emulator emulates each of said ROM banks with a different RAM page.

41. (Previously presented) The method of claim 6 further including using a ROM selection pointer to select a current ROM page and a ROM page count register to specify the number of emulated ROM pages that have been loaded.

42. (Previously presented) The method of claim 6 further including using a no-write functional module to protect allocated emulated ROM space so that inadvertent write instructions do not succeed in overwriting emulated read only memory.

43. (Previously presented) The method of claim 6 further including using a no-write function to protect emulated ROM space from being written to,

thus making sure the emulated ROM space is emulated as read only memory rather than read-write memory.

44. (Previously presented) The method of claim 6 further including using function pointers to implement no-write allocated ROM space protection.

Claims 45-58 (Canceled)